

### **Amendments to the Claims**

Please amend the claims as follows:

1. (Currently amended) In a computing system having a processor storing a software product having program steps that controls said system for providing multi-threaded programming support, said system comprising:

a thread monitor class providing thread monitoring services to threads of a multi-threaded process, the thread monitor class including:

a thread registration method that allows a thread to optionally self register itself ~~a thread~~ for monitoring by the class; and

a thread monitoring supervisor to monitor all threads registered for monitoring operation of threads that invoke the thread registration method.

2. (Currently amended) The system of claim 1 wherein the thread monitor class further includes:

a thread un-registration method that allows a thread to optionally remove itself ~~from~~ a prior registration of itself ~~of a thread~~ for monitoring by the class.

3. (Previously Presented) The system of claim 1 wherein the thread monitor class further includes:

a stop thread monitoring method to optionally terminate monitoring of all threads registered for monitoring by the class.

4. (Currently amended) The system of claim 1 wherein the thread monitor class further includes:

a thread Heart Beat method to signal a Heart Beat from a thread that optionally self-registers itself ~~registered~~ for Heart Beat monitoring by the class.

5. (Currently amended) The system of claim 1 wherein the thread registration method comprises:

a thread alive check registration method that allows a thread to optionally self register itself ~~invoked by a thread to register~~ for monitoring by the class wherein the monitoring comprises periodically verifying that the invoking thread is still alive.

6. (Currently amended) The system of claim 1 wherein the thread registration method comprises:

a thread poll registration method that allows a thread to optionally self register itself ~~invoked by a thread to register~~ for monitoring by the class wherein the monitoring comprises periodically verifying that the ~~invoking~~ self registered thread is properly operating by invoking a poll method derived from the thread poll registration invocation; and

said derived poll verifies the functionality of the ~~threaded registration~~ self registered thread.

7. (Currently amended) The system of claim 1 wherein the thread registration method comprises:

a thread Heart Beat registration method that allows a thread to optionally self register itself ~~invoked by a thread to register~~ for monitoring by the class wherein the monitoring comprises periodically verifying that the ~~invoking~~ self-registered thread is still alive and not hung based on receipt of periodic Heart Beat method invocations from the thread invoking the thread Heart Beat registration method.

8. (Previously Presented) The system of claim 1 wherein the thread monitoring supervisor is optionally instantiated within a main thread of a multi-threaded program.

9. (Previously Presented) The system of claim 1 wherein the thread monitoring supervisor is further optionally operable to restart an inoperable thread.

10. (Previously Presented) The system of claim 1 wherein the thread monitoring supervisor is further optionally operable to restart the process that includes an inoperable thread.

11. (Currently amended) A method comprising a software product in a data processing system for monitoring operability of multiple threads: said method comprising the software product for controlling the data processing steps of:

Instantiating a thread monitoring supervisor in a thread of a multi-threaded process;

optionally self registering one or more additional threads of the multi-threaded process for the monitoring of their operation by the thread monitoring supervisor; and

monitoring the operability of [[the]] additional self registered "threads" by the operation of the thread monitoring supervisor.

12. (Currently amended) The method of claim 11

wherein the step of self registering further comprises optionally registering [[the]] an additional thread as a Heart Beat thread for monitoring according to Heart Beat signals,

wherein said additional thread is operable to periodically communicate a Heart Beat signal with the monitoring supervisor, and

wherein the step of monitoring further comprises detecting periodic receipt of Heart Beat signals to monitor operability of said additional self registered thread.

13. (Currently Amended) The method of claim 11 wherein the step of monitoring further comprises determining whether said additional self registered thread is still alive to monitor operability of said additional self registered thread.

14. (Currently amended) The method of claim 11 wherein the step of registering further comprises optionally self registering the additional thread as a polling thread associated with a poll function to indicate the operability status of the additional self registered thread, and

wherein the step of monitoring further comprises periodically invoking the ~~poll~~ polling function associated with the additional self registered thread to monitor operability of the additional self registered thread.

15. (Previously Presented) The method of claim 11 wherein the step of instantiating further comprises optionally instantiating the thread monitoring supervisor in a main thread of the multi-threaded process.

16. (Previously Presented) The method of claim 11 further comprising optionally restarting an inoperable thread.

17. (Previously Presented) The method of claim 11 further comprising optionally restarting a process that includes an inoperable thread.

18 (Previously Presented) The system of claim 1 wherein the thread monitor is a generic and reusable component.